Computing and Informatics, Vol. 30, 2011, 1067–1081

RELIABLE AND EFFICIENT WAY TO BROADCAST MESSAGES IN A GROUP BY TRUST-BASED BROADCAST (TBB) SCHEME

Ailixier Aikebaier, Tomoya Enokido, Makoto Takizawa

Department of Computers and Information Science
Faculty of Science and Technology
Seikei University
3-3-1 Kichijoji-kitamachi, Musashino-shi
Tokyo 180-8633, Japan
e-mail: {alisher.akber, makoto.takizawa}@computer.org, eno@ris.ac.jp

Abstract. Nowadays information systems are being shifted to distributed architectures, i.e. Grid and Peer-to-peer (P2P) models to obtain the benefits like scalability, autonomy, and fault-tolerance. We consider the P2P model as a fully distributed, scalable system, which is composed of peer processes (peers). Here, a group of multiple peers cooperate with each other. Peers have to efficiently and flexibly deliver messages to every peer of the group in P2P overlay networks. In order to efficiently and reliably broadcast messages in a scalable group, we take advantage of the multipoint relaying (MPR) mechanism. Here, each peer sends messages to only a subset of its acquaintances. However, if a peer which forwards messages to other peers is faulty, the peers cannot receive messages. In this paper, we newly discuss a trustworthiness-based broadcast (TBB) algorithm where only trustworthy peers forward messages. That is, untrustworthy peers are peers which cannot forward the messages due to some faults. Here, the transmission fault implied by faults of untrustworthy peers can be reduced. We evaluate the TBB algorithm in terms of the number of messages transmitted.

Keywords: Distributed systems, trustworthiness, message broadcasting

Mathematics Subject Classification 2000: 6806

1 INTRODUCTION

In the scalable peer-to-peer (P2P) model [26, 18], every process is peer and there is no centralized coordinator. In P2P applications like Intelligent Decision Advisor (IDA), Distributed Decision Making (DDM), and Computer Supported Cooperative Work (CSCW), a group of multiple peers are required to do cooperation, for example, to fix a date of a meeting and to find a best location to build a building.

In group communications [14, 15], each peer sends messages to every peer and receives messages from every peer in a group. There are many discussions on how to causally deliver messages in a group [19]. Efficient mechanisms to broadcast messages to every peer are required in order to casually deliver messages in a scalable group. The basic approach to broadcasting messages is represented by the flooding algorithms [22]. Here, each peer sends a message to its acquaintances and the acquaintances forward the messages to their neighbour peers. However, more messages are transmitted.

In order to reduce the number of messages, we take advantage of the *multipoint relying* (*MPR*) mechanism [21]. Here, each peer transmits a message to every neighbor peer but only some, not all of the neighbor peers forward the message to their neighbor peers. However, we have to sacrifice some level of reliability of the system since a message is not forwarded if peers to forward the message are faulty. In order to increase the fault-tolerance, we introduce the trustworthiness concept of a peer, which shows how much another peer trusts the peer with respect to the message forwarding. We discuss a novel *trustworthiness-based broadcast* (*TBB*) algorithm to reliably and efficiently deliver messages to every peer in a group. Here, each peer sends a message to its neighbor peers and only trustworthy peers out of the neighbor peers forward the message to their neighbors. In this paper, untrustworthy peers are assumed to be peers which only receive messages but cannot forward the messages due to faults. Hence, even if untrustworthy peers are faulty, other peers can receive messages through trustworthy peers. We evaluate the TBB algorithm to broadcast messages in a group in terms of number of messages transmitted compared with the MPR and traditional flooding algorithms. We show the number of messages can be reduced and the messages are more reliably delivered to every peers in the TBB algorithm compared with the traditional MPR algorithm with no trustworthiness concept involved in the algorithm.

In Section 2, we briefly present the multipoint relay (MPR) mechanism. In Section 3, we discuss the trustworthiness concept and the TBB algorithm. In Section 4, we evaluate the TBB algorithm in terms of number of messages and reliability of message delivery compared with the MPR and flooding schemes.

2 MULTIPOINT RELAYING (MPR) SCHEME

2.1 Basic Algorithm

In a group of multiple peers, each peer has to deliver a message to all the other peers. In a scalable P2P overlay network, each peer cannot directly send a message to every other peer of a group. Each peer can only send a message to its neighbor peers, i.e. *acquaintance* peers [27]. In one approach to broadcasting a message, a peer p_i first sends a message to every neighbor peer p_j. Upon receipt of a message, the peer p_j forwards the message to its neighbor peers. This is a pure flooding scheme [22] where messages are forwarded from peer to their neighbor peers. However, the pure flooding scheme implies the huge network overhead due to the message explosion.

The concept of "multipoint relaying (MPR)" scheme is developed to reduce the number of duplicate transmissions while each peer forwards a message to the neighbor peers [21]. Here, on receipt of a message, a peer forwards the message to all the neighbor peers but only some of the neighbor peers forward the message to other peers. Each peer is assumed to know not only the first neighbor peers but also the second neighbor peers. First neighbor peers are peers with which the peer p_i can communicate directly. The peer p_i is assumed to know every second neighbor peer, but cannot directly communicate with it. By taking into consideration the second neighbor peers in addition to the first neighbor peers, each peer selects a subset of the first neighbor peers only which forward the message. The selected neighbor peers are referred to as *relay* peers. The other neighbor peers which just receive the message and do not forward the message are *leaf* peers. Since the number of messages transmitted can be reduced, the MPR scheme provides an adequate solution to reduce the overhead to broadcast messages in P2P overlay networks. Every leaf peer just receives a message from a relay peer while every relay peer forwards the message to the neighbor peers.

Let $N(p_i)$ be a set of first neighbor peers of a peer p_i. A set of the second neighbor peers of a peer p_i is denoted by $N^2(p_i)$. $N^2(p_i) = \cup_{p_j \in N(p_i)} N(p_j) - N(p_i)$. Let $R(p_i)$ and $L(p_i)$ be collections of replay peers and leaf peers of a peer p_i, respectively. Here, $N(p_i) = R(p_i) \cup L(p_i)$, $R(p_i) \cap L(p_i) = \phi$, and $N^2(p_i) = \cup_{p_j \in R(p_i)} N(p_j)$. That is, a message sent by a peer p_i can be delivered to every second neighbor peer of p_i where only the relay peers of p_i forward the message to second neighbor peers of p_i. A peer p_j is referred to as *covered* by a peer p_i iff $p_j \in N(p_i)$ or p_j is covered by some relay peer $p_k \in R(p_i)$. A collection of peers covered by a peer p_i is referred to as subnetwork *covered* by the peer p_i. $S(p_i)$ shows a subnetwork of the peers covered by a peer p_i. Here, the peer p_i is a *root* of the subnetwork $S(p_i)$. A peer p_k in $S(p_i) \cap S(p_j)$ is *redundantly* covered by a pair of peers p_i and p_j. If a peer p_k is covered by only one peer p_i, the peer p_k is referred to as *simply covered*, i.e. $p_k \in S(p_i)$ but $p_k \notin S(p_j)$ for every $p_j (\neq p_i)$. Suppose a peer p_k is simply covered by a peer p_i. If the peer p_i does not forward a message, the peer p_k does not receive the message. If the peer p_k is redundantly covered by not only the peer p_i but also p_j, p_k can receive a message through p_j even if p_i does not forward the message.

An algorithm $MPR(p_i, N(p_i))$ for selecting a set $R(p_i)$ of relay peers [21] in $N(p_i)$ is shown as follows:
[**MPR**$(p_i, C(p_i)$] A collection $R(p_i)$ of relay peers are selected in $C(p_i)$ and each relay peer p_j in $R(p_i)$ is assigned with a set $C(p_j)$.

1. Start with an empty multipoint relay set $R(p_i)$;

$$R(p_i) = \phi, \qquad S = N^2(p_i), \qquad F = C(p_i).$$

2. While $F \neq \phi$, do the following steps:
   (a) select a neighbor peer p_j in F where $N(p_j) \cap N(p_k) = \phi$ for every other first neighbor peer p_k in F
   (b) if found,

$$R(p_i) = R(p_i) \cup \{p_j\}, \qquad S = S - N(p_j), \qquad F = F - \{p_j\}$$

   (c) if not found, go to step 3.
3. If $F = \phi$, end:
4. While $S \neq \phi$, do the following steps:
   (a) for each peer p_j in F, obtain a subset $U(p_j)$ of peers which p_j covers in the set S, $U(p_j) = N(p_j) \cap S$
   (b) select a peer p_j where $|U(p_j)|$ is the maximum,

$$R(p_i) = R(p_i) \cup \{p_j\}, \qquad S = S - U(p_j), \qquad F = F - \{p_j\}, \qquad C(p_j) = U(p_j).$$

5. For each peer p_j in F, $C(p_j) = \phi$, i.e. p_j is a leaf peer.
6. For each relay peer p_j in $R(p_i)$, $MPR(p_j, C(p_j))$.

In the MPR algorithm, for each neighbor peer p_j in $N(p_i)$, $C(p_j)$ is obtained as a set of neighbor peers of p_j. Here, the peer p_i is a *parent* of p_j and p_j is a *child* of p_i. If p_j is a leaf peer, $C(p_j) = \phi$. For each neighbor peer p_j in $C(p_i)$, the algorithm MPR is recursively applied to obtain a set $R(p_j)$ of relay peers of p_j. In the MPR algorithm, an *directed acyclic graph* (DAG) is obtained.

2.2 Faults

In a DAG, a parent node p_i of a peer p_j shows a relay peer which forwards messages to the child peer p_j upon receipt of the messages. A collection of the child peers of a peer p_i is shown as $C(p_i)$. Let $R(p_i)$ indicate a set of relay peers of a peer p_i obtained by the MPR algorithm. $U(p_i)$ is a set of leaf peers of a peer p_i. Peers colored black and white show relay and leaf peers, respectively in Figure 1.

A relay peer plays a critical role to broadcast messages in a group. If a relay peer p_i is faulty, every peer simply covered by the faulty peer p_i is not able to receive

messages which are sent to the peer p_i. Let us consider a subnetwork S ($= C(p)$) covered by a peer p shown in Figure 1, which is circled by the line. A peer p is a root of the subnetwork S. The peers a, b, c, and d in S are simply covered by the peer p. Suppose the peer p is faulty. Here, every peer in S cannot receive messages sent to the peer p. Thus, if a relay peer p_i is faulty, peers in a sub-network of the peer p_i may not receive messages.

In order to increase the reliability for broadcasting messages, we newly introduce the trustworthiness of a neighbor peer. A *trustworthy* peer is a peer which receives a message m and forwards only the correcte message m to its neighbor peers. An *untrustworthy* peer is a peer which receives a message m but may not forward the message m or may send an incorrect message m' ($\neq m$) to its neighbor peers. Hence, if a peer p_i is simply covered by an untrustworthy peer p_j, the peer p_i may not be able to receive a corrected message. Hence, a peer p_i selects trustworthy neighbor peers as relay peers. Then, the peer p_i sends a message to the neighbor peers and only the trustworthy neighbor peers forward the message to their neighbor peers.

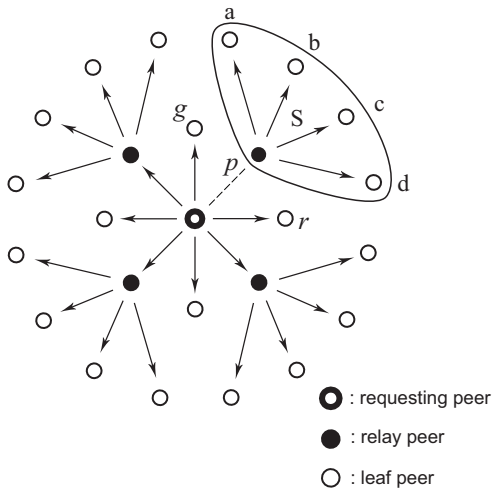

Fig. 1. Failure in multipoint relays

Let us consider Figure 2 a) as an example. Here, let $T(p_i)$ show the trustworthiness value of a peer p_i. In Figure 2, suppose $T(g) > T(r) > T(p)$ for three peers g, r and p. Here, we select the most trustworthy peer g as a relay peer. Then, the peer g forwards a message to every peer in the subnetwork S. This is an ideal case, that is, the subnetwork S which is originally covered by the peer p can be also covered by the peer g. However, the peer g might not be able to cover every peer as shown in Figure 2 b). Therefore, another peer has to be selected to cover the peers which the peer g does not cover. In Figure 2 b), the peers c and d uncovered by the peer g are covered by the second most trustworthy peer r. The overall idea is

that every subnetwork is covered by a most trustworthy relay peer. It depends on the overlay topology among peers how many relay peers are required to cover all the peers in a subnetwork. In Figure 2 b), one more relay peer is required to cover the same subnetwork S as shown in Figure 1. If we use more trustworthy neighbor peers to transmit messages to others, we can increase the overall fault-tolerance of the multipoint-relay mechanism.

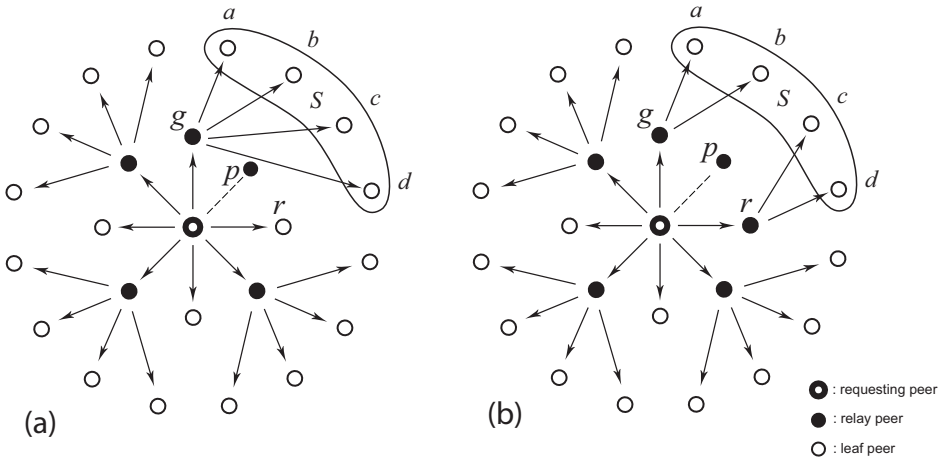

Fig. 2. Trusted neighbors in multipoint relays

3 TRUSTWORTHINESS-BASED BROADCAST (TBB) SCHEME

3.1 Peer Trustworthiness

P2P systems are scalably composed of multiple peers in a decentralized manner. This means, each peer has to obtain information of other peers and propagate the information to other peers through neighbor peers. A neighbor peer p_j of a peer p_i means that p_i can directly communicate with p_j. Thus, it is significant for each peer to have some number of neighbor peers. Moreover, it is more significant to discuss if each peer has trustworthy neighbor peers. In reality, each peer might be faulty. If some peer p_j is faulty, other peers covered by p_j may not receive messages. It is critical to discuss how a peer can trust each of its neighbor peers [27]. In this paper, we newly introduce a trustworthiness based algorithm to broadcast messages in a scalable group, by which messages can be more reliably and efficiently broadcast to every peer.

Suppose a source peer p_r would like to broadcast a message m in a group. The peer p_r selects a neighbor peer p_i as a relay peer for broadcasting the message m to the other peers. Let $T_r(p_i)$ show the trustworthiness of a neighbor peer p_i of

a peer p_r, which the peer p_r holds. $N(p_r)$ shows a collection of neighbor peers of the peer p_r. The peer p_r calculates the trustworthiness $T_r(p_i)$ for each neighbor peer p_i by collecting the trustworthiness values $T_k(p_i)$ on the peer p_i from every neighbor peer p_k in $N(p_r)$ which can communicate with both p_i and p_r, i.e. $p_k \in N(p_r) \cap N(p_i)$. There is some possibility that the peer p_i is faulty or sends incorrect information. Hence, the peer p_r does not consider the trustworthiness $T_i(p_i)$ from the target peer p_i to calculate the trustworthiness $T_r(p_i)$.

A peer p_k sends a request to the peer p_i and receives a reply from p_i. This request-reply interaction is referred to as *transaction*. If the peer p_k receives a successful reply, the transaction is successful; otherwise it is unsuccessful. The peer p_k considers the neighbor peer p_i to be more trustworthy if p_k had more successful transactions for p_i. Let $ST_k(p_i)$ indicate the *subjective* trustworthiness $T_k(p_i)$ on the target peer p_i which a peer p_k obtains through directly communicating with the peer p_i. Let $tT_k(p_i)$ show the total number of transactions which p_k issues to p_i. Let $sT_k(p_i)$ ($\leq tT_k(p_i)$) be the number of successful transactions from p_k to p_i. Here, the subjective trustworthiness $ST_k(p_i)$ is calculated as follows:

$$ST_k(p_i) = \frac{sT_k(p_i)}{tT_k(p_i)}. \tag{1}$$

If the peer p_i is not a neighbor peer p_k, $p_i \notin N(p_k)$, the peer p_k does obtain the subjective trustworthiness $ST_k(p_i)$. In addition, if the peer p_k had not issued any transaction to the peer p_i even if $p_i \in N(p_k)$, i.e. $tT_k(p_i) = 0$, $ST_k(p_i) = \perp$ (not defined). Thus, through communicating with each neighbor peer p_k, each peer p_r obtains the subject trustworthiness $ST_k(p_i)$ for the neighbor peer p_i. The subjective trustworthiness $ST_k(p_i)$ shows how reliably a peer p_i is recognized by a peer p_k. Therefore, if a peer p_r would like to get the trustworthiness of a target peer p_i, the peer p_r asks each neighbor peer p_k to send the subjective trustworthiness $ST_k(p_i)$. Each neighbor peer p_k keeps in record the subject trustworthiness $ST_k(p_i)$ in the log. Here, let $TN(p_r)$ be a collection of neighbor peers which send the non-null subjective trustworthiness $ST_k(p_i)$ to the peer p_r. After collecting the subjective trustworthiness $ST_k(p_i)$ from each neighbor peer p_k, the source peer p_r calculates the trustworthiness $T_r(p_i)$ on the neighbor peer p_i by the following equation:

$$T_r(p_i) = \frac{\sum_{p_k \in TN(p_r) - \{p_i\}} ST_k(p_i)}{|TN(p_r) - \{p_i\}|} \tag{2}$$

Let us consider peers shown in Figure 3. Here, a source peer p_r would like to know the trustworthiness $T_r(p_i)$ of a neighbor peer p_i. The peer p_r has five neighbor peers, p_1, p_2, p_3, p_4, and p_i. Here, $N(p_r) = \{p_1, p_2, p_3, p_4, p_i\}$. The peer p_i is excluded from $N(p_r)$ since p_i is a target peer, i.e. $S = N(p_r) - \{p_i\} = \{p_1, p_2, p_3, p_4\}$. Here, the source peer p_r requests each neighbor peer p_k in the neighbor set S to send the subjective trustworthiness $ST_k(p_i)$ $(k = 1, 2, 3, 4)$. After receiving the subjective trustworthiness of the peer p_i from all the four neighbors, the peer p_r calculates the

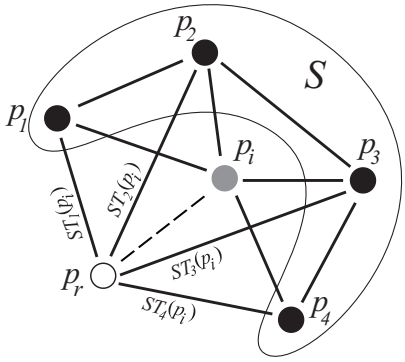

Fig. 3. Peer trustworthiness

trustworthiness $T(p_i)$ by using the Equation (2), i.e. $T_r(p_i) = (ST_1(p_i) + ST_2(p_i) + ST_3(p_i) + ST_4(p_i))/4$.

3.2 Trustworthiness-Based Broadcast (TBB) Algorithm

By taking advantage of the trustworthiness concept of each neighbor peer, the original multipoint relay (MPR) algorithm is modified as the trustworthy-based broadcast (TBB) scheme. In order to select relay peers of a peer p_r, the algorithm $TBB(p_r, C(p_r))$ is applied where $C(p_r)$ is collection of the first neighbor peers of p_r. $\mathbf{TBB(p_r, C(p_r))}$

1. Start with an empty multipoint relay set $MR(p_r), MR(p_r) = \phi$. $S = N^2(p_r)$, $F = C(p_r)$. Let TF be a set of trustworthy neighbors of p_i, i.e. $\{p_j \in C(p_r) | T_r(p_j) \geq \alpha_r\}$ where $0 \leq \alpha_r \leq 1$. α_r gives a threshold value on the trustworthiness. If $T_r(p_i)$ is larger than or equal to α_r, the peer p_r recognizes the neighbor peer p_i to be trustworthy. Otherwise, p_i is considered to be untrustworthy.
2. While $TF \neq \phi$, do the following steps:
   (a) select a trustworthy neighbor peer p_i in TF such that $N(p_i) \cap N(p_j) = \phi$ for every trustworthy peer p_j in TF $(p_j \neq p_i)$
   (b) if found, $F = F - \{p_i\}$, $TF = TF - \{p_i\}$, $S = S - N(p_i)$, $MR(p_r) = MR(p_r) \cup \{p_i\}$
   (c) if not found, go to step 3.
3. While $TF \neq \phi$, do the following steps:
   (a) $U(p_j) = N(p_j) \cap S$ for each p_j in TF
   (b) select a trustworthy neighbor peer p_i in TF such that $|U(p_i)|$ is the maximum, i.e. the number of neighbor peers which are not covered is the maximum

(c) $F = F - \{p_i\}$, $TF = TF - \{p_i\}$, $SS = S$, $S = S - N(p_i)$, $MR(p_r) = MR(p_r) \cup \{p_i\}$, $C(p_i) = N(p_i) \cap SS$.

4. While $F \neq \phi$, $TF = \phi$ do the following steps:

   (a) select a peer p_j in F such that $|N(p_j) \cap S|$ is the minimum
   (b) $F = F - \{p_j\}$, $SS = S$, $S = S - N(p_j)$, $MR(p_r) = MR(p_r) \cup \{p_j\}$, $C(p_i) = N(p_i) \cap SS$.

5. For each relay neighbor peer p_i in $MR(p_r)$, $TBB(p_i, C(p_i))$.

For each neighbor peer p_i, $C(p_i)$ gives a collection of neighbor peers to which p_i forwards a message, $C(p_i) \subseteq N(p_i)$. If p_i is not a relay peer, $C(p_i) = \phi$. Otherwise, $C(p_i) = MR(p_i) \cup U(p_i)$ and $MR(p_i) \cap U(p_i) = \phi$. In step 4, each untrustworthy neighbor peer p_i is assigned with as small number of neighbors as possible. Even if p_i is faulty, a smaller number of peers are effected.

Let $MT(p_r)$ be a directed acyclic graph (DAG) of a peer p_r obtained by the algorithm $TBB(p_r, N(p_r))$. The *DAG* $MT(p_i)$ is *trustworthy* if every non-leaf, relay peer is trustworthy. In a trustworthy *DAG*, every untrustworthy peer is a leaf peer. Hence, even if an untrustworthy peer p_i does not forward a meesage, every trustworthy peer can reciefve the messages.

In this paper, a faulty peer is assumed to receive a message but is not able to forward the message to other peers. An algorithm is referred to as *sound* iff a message can be delivered to all the peers in the network.

4 EVALUATION

Compared with the original MPR algorithm and pure flooding (PF) algorithm, we evaluate the proposed trustworthiness-based broadcast (TBB) algorithm in terms of the number of messages transmitted to broadcast a message in group of n peers. In this evaluation, we consider an $L \times L$ grid structured overlay network. The total number n of peers in the network is L^2. We also assume that each peer has an identifier (ID). Since both of the MPR algorithm and the TBB algorithm aim at reducing the number of messages unnecessarily transmitted, we measure the number of messages which are sent in each algorithm. As we mentioned in the preceding section, if a relay peer p_i is faulty, peers simply covered by p_i cannot receive messages. Hence, we evaluate the TBB, MPR, and PF algorithms in presence of faulty peers.

In the evaluation, some number of peers are randomly selected to be faulty out of the n peers. F shows the ratio of the faulty peers to the total number n of peers in the network. For example, "$F = 0.05$" means that five percent of the peers are faulty. T_i shows the trustworthiness of a peer p_i. T_i is a value randomly chosen in the range of 0.1 to 1.0. The higher T_i is, the more trustworthy the peer p_i is. First, the trustworthiness T_i is given to each peer p_i. Then, based on the faulty ratio F, faulty peers are selected. Depending on the trustworthiness value T_i of each peer p_i,

we select a peer which has the smallest T_i value to be faulty. If we found multiple peers which have the same lowest T_i value, we take a peer whose peer ID is the highest. That is, the lower trustworthiness T_i a peer p_i has, the more likely p_i will be faulty.

We evaluate the TBB, MPR, and PF algorithms for different faulty ratios F in the network. Figures 4 and 5 show the numbers of messages with total number n of peers for $F = 0.05$ and $F = 0.1$, respectively. Here, for $F = 0$ and $F = 0.05$, a message can be delivered to all the peers by using fewer number of messages in the MPR algorithm than the TBB algorithm. In the pure flooding scheme, the largest number of messages are transmitted to deliver messages as shown in Figures 4 and 5. However, if ten percentages of the peers are faulty in the network ($F = 0.1$), a message cannot be delivered to all the peers in the MPR algorithm, i.e. MPR is not sound. On the other hand, the TBB algorithm is sound, i.e. a message can be delivered to all the peers with fewer number of messages than the pure flooding as shown in Figure 5. Thus, the TBB algorithm is more sound, i.e. more reliable and more efficient, i.e. fewer number of messages are transmitted.

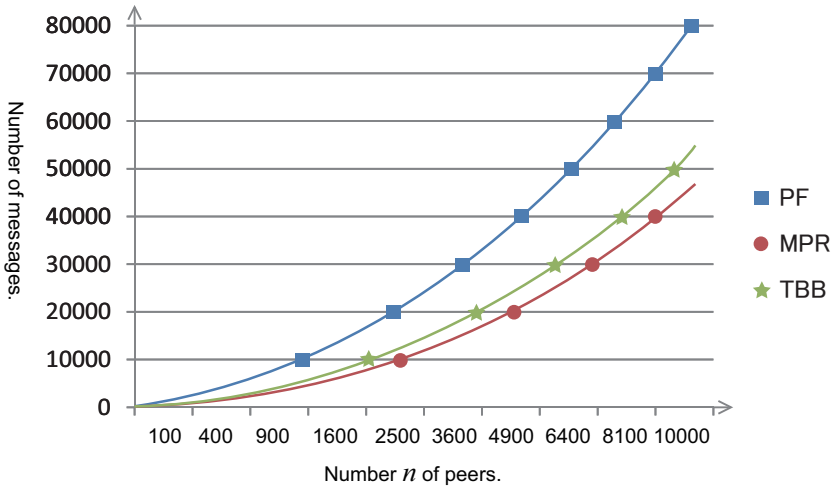

Fig. 4. Number of messages ($F = 0.05$)

Figure 6 shows the average value of network coverage of each algorithms to the faulty ratio F of the network where the number n of peers is taken from 100 to 10 000. In the MPR algorithm, messages cannot be delivered to all the peers for larger than about six percentages of faulty peers ($F = 0.06$). For $F = 0.1$, about 40 percentage of the peers cannot receive messages. On the other hand, in the TBB algorithm, messages cannot be delivered to all the peers for $F > 0.18$. For $F = 0.27$, more than 90 percentages of the peers can receive messages. Figure 7 shows the average value of number of messages for the faulty ratio F where is $100 \leq n \leq 10\,000$. As shown

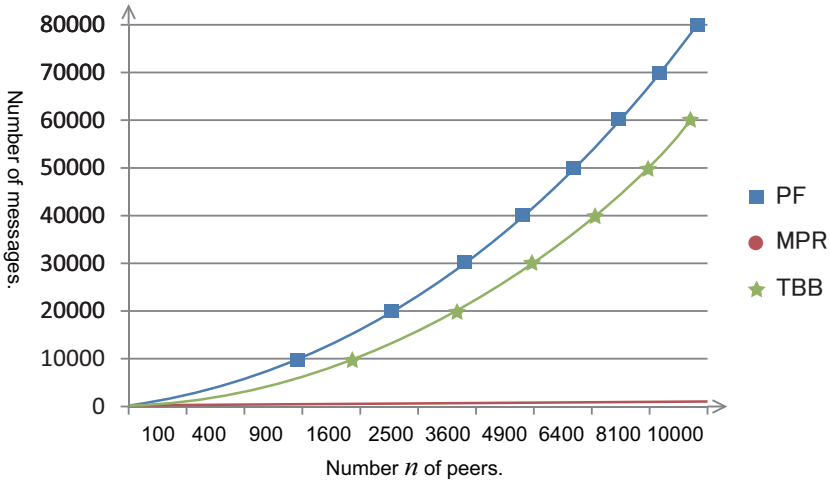

Fig. 5. Number of messages ($F = 0.1$)

in Figure 7, the TBB algorithm can cover the same network with less messages than the PF and MPR ones. In addition, in reality, the situation like about 20 percent of the peers are faulty in a network is unlikely.

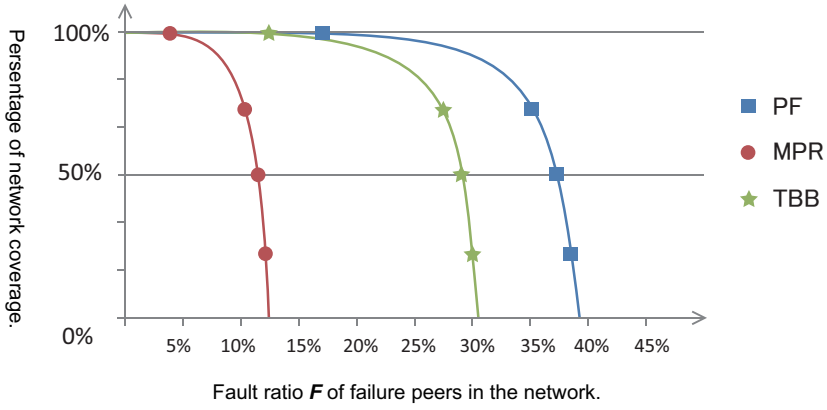

Fig. 6. Network coverage to fault ratio

5 CONCLUDING REMARKS

We discussed how to efficiently and reliably broadcast messages to all the peers in a scalable group. We introduced the novel trustworthiness concept of neighbor peers and discussed the trustworthiness-based broadcast (TBB) algorithm to broadcast

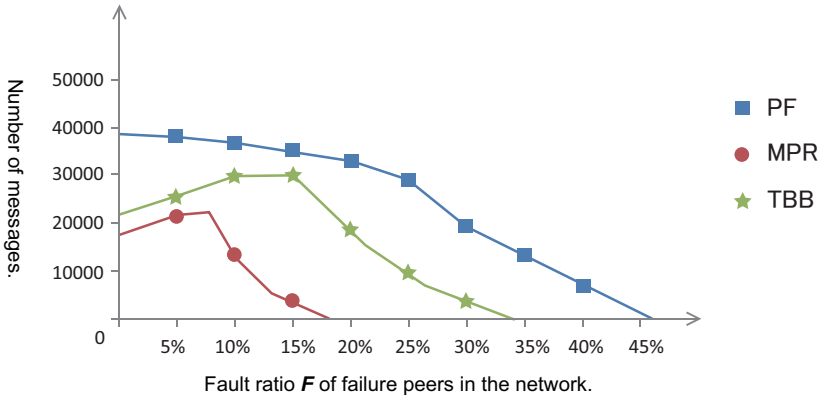

Fig. 7. Number of messages to fault ratio

messages. By making trustworthy peers forward messages to other peers, we can remove the effect of faulty peers to deliver messages to all the peers. The evaluation results show that, in the network where the 10 percent of peers are faulty, the original MPR algorithm is not sound, i.e. unable to deliver a message to all the peers in the network while the TBB algorithm can still deliver the message to all the peers. Thus, messages can be efficiently and reliably delivered to all the peers in the TBB algorithm.

Acknowledgments

This research is supported by Research Fellowships of Japan Society for the Promotion of Science for Young Scientists (JSPS). This research was also partially supported by the strategy research project of Seikei University and MEXT, Grant in Aid for Building Strategy Research Infrastructure.

REFERENCES

[1] Aikebaier, A.—Enokido, T.—Takizawa, M.: Checkpointing in a Distributed Coordination Protocol for Multiple Peer Processes. In: Proc. of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008), pp. 48–54.

[2] Aikebaier, A.—Hayashibara, N.—Enokido, T.—Takizawa, M.: A Distributed Coordination Protocol for a Heterogeneous Group of Peer Processes. In: Proc. of the IEEE 21th Conference on Advanced Information Networking and Applications (AINA 2007), pp. 565–572.

[3] Aikebaier, A.—Hayashibara, N.—Enokido, T.—Takizawa, M.: Making an Agreement in an Order-Heterogeneous Group by Using a Distributed Coordina-

tion Protocol. In: Proc. of the 2nd International Workshop on Advanced Distributed and Parallel Network Applications (ADPNA 2007), CD-ROM.

[4] Aikebaier, A.—Enokido, T.—Takizawa, M.: A Distributed Coordination Protocol for Multiple Peer Processes. In: Proc. of IEEE the 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), CD-ROM.

[5] Aikebaier, A.—Barolli, V.—Enokido, T.—TAKIZAWA, M.: Recoverable Cuts to Make Agreement among Peers. In: Proc. of IEEE the 23nd International Conference on Advanced Information Networking and Applications (AINA-2009), CD-ROM.

[6] Aikebaier, A.—Enokido, T.—Takizawa, M.: Efficiently Making Agreement among Peer Processes by using Recoverable Cuts. In: Proc. of the 3rd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009), CD-ROM.

[7] Aikebaier, A.—Enokido, T.—Takizawa, M.: Trustworthiness Among Peer Processes in Distributed Agreement Protocol. In: Proc. of IEEE 24nd International Conference on Advanced Information Networking and Applications (AINA 2010), CD-ROM.

[8] CORMAN, A. B.—Schachte, P.—Teague, V.: A Secure Group Agreement (SGA) Protocol for Peer-to-Peer Applications. In: Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW07), pp. 24–29.

[9] Ezhilchelvan, P.—Morgan, G.: A Dependable Distributed Auction System: Architecture and an Implementation Framework. In: Proc. of the IEEE 5th International Symposium on Autonomous Decentralized Systems (ISADS), pp. 3–7.

[10] Gray, J.—Lamport, L.: Consensus on Transaction Commit. ACM Transactions on Database Systems (TODS) archive, Vol. 31, 2006, No. 1, pp. 133–160.

[11] Hayes, B.: Cloud Computing. Communications of the ACM, Vol. 51, 2008, No. 7, pp. 9–11.

[12] Hurfin, M.—Raynal, M.—Tronel, F.—Macedo, R.: A General Framework to Solve Agreement Problems. In: Proc. of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS), 1999, pp. 56–65.

[13] Kling, R.: Cooperation, Coordination and Control in Computer-supported Work. Communications of the ACM, Vol. 34, 1991, No. 12, pp. 83–88.

[14] Lamport, L.: Time, Clocks and the Ordering of Events in a Distributed System. Communications of the ACM 21, Vol. 7, 1978, pp. 558–565.

[15] Chockler, G. V.—Keidar, I.—Vitenberg, R.: Group Communication Specifications: A Comprehensive Study. ACM Computing Surveys (CSUR), Vol. 33, 2001, No. 4, pp. 427–469.

[16] Lamport, L.—Shostak, R.—Pease, M.: The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems, Vol. 4, 1982, No. 3, pp. 382–401.

[17] Lee, P.—Lui, J.—Yau, D.: Distributed Collaborative Key Agreement Protocols for Dynamic Peer Groups. In: Proc. of the 10th IEEE International Conference on Network Protocols, 2002, pp. 322–331.

[18] Montresor, A.: A Robust Protocol for Building Superpeer Overlay Topologies. In: Proc. of the 4th International Conference on Peer-to-Peer Computing, 2004, pp. 202–209.

[19] Kawanami, S.—Enokido, T.—Takizawa, M.: A Group Communication Protocol for Scalable Causal Ordering. In: Proc. of the 18th International Conference on Advanced Information Networking and Applications (AINA '04), Vol. 1, pp. 296–301.

[20] Napster website, `http://www.napster.com`.

[21] Qayyum, A.—Viennot, L.—Laouiti, A.: Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. In: Proc. of the 35th Annual Hawaii International Conference on System Sciences, 2002, pp. 3866–3875.

[22] Ripeanu, M.—Foster, I.: Mapping Gnutella Network. IEEE Internet Computing, January/February 2002, pp. 50–57.

[23] Sabater, J.—Sierra, C.: Reputation and Social Network Analysis in Multi-Agent Systems. In: Proc. of the First International Joint Conference on Autonomous Agents and Multiagent Systems 2002, Part 1, pp. 475–482.

[24] Shimojo, I.—Tachikawa, T.—Takizawa, M.: M-ary Commitment Protocol with Partially Ordered Domain. In: Proc. of the 8th International Conference on Database and Expert Systems Applications (DEXA), Vol. 197, pp. 397–408.

[25] Skeen, D.: NonBlocking Commit Protocols. Proc. of the ACMSIGMOD International Conference on Management of Data 1981, pp. 133–142.

[26] Upadrashta, Y.—Vassileva, J.—Grassmann, W.: Social Networks in Peer-to-Peer Systems. In: Proc. of the 38th Hawaii International Conference on System Sciences (HICSS-38 2005), CD-ROM.

[27] Watanabe, K.—Nakajima, Y.—Enokido, T.—Takizawa, M.: Ranking Factors in Peer-to-Peer Overlay Networks. ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 2, 2007, No. 3, Article No. 11.

Ailixier Aikebaier received his Bc. E. Degree in Computers and Systems Engineering from XinJiang University, China, in 2000, and M. E. Degree in Computers and Systems Engineering from Tokyo Denki University, Japan in 2009. He is currently a Ph. D. candidate student in graduate school of Science and Technology, Seikei University, Japan. He won the best paper award at CISIS2008 and CISIS2010. His research interests include distributed systems, P2P networks, consensus problems and fault-tolerant systems.

Tomoya Enokido received Bc. E. and M. E. Degrees in Computers and Systems Engineering from Tokyo Denki University, Japan in 1997 and 1999, respectively. After that he worked for NTT Data Corporation. He joined Tokyo Denki University in 2002. He received his D. E. Degree in Computer Science from Tokyo Denki University in 2003. After that he worked for Computers and Systems Engineering as a research associate. He joined Faculty of Business Administration of Rissho University in 2005. He is an Associate Professor in the Faculty of Business Administration, Rissho University. His research interests include distributed systems. He is a member of IEEE.

Makoto Takizawa is a Professor in the Department of Computer and Information Science, Seikei University. He was a Professor and the Dean of the Graduate School of Science and Engineering, Tokyo Denki University. He was a Visiting Professor at GMD-IPSI, Keele University, and Xidian University. He was one of the Board of Governors, is an IEEE CS Golden Core member and an IPSJ fellow. He received his DE in Computer Science from Tohoku University. He chaired many international conferences such as IEEE ICDCS, ICPADS, and DEXA. He founded IEEE AINA. His research interests include distributed systems and computer networks.